Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
6 6	1 2 0 1		
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14

Termin: Mittwoch, 24. November 2021

IHK

Abschlussprüfung Winter 2021/22

1201

1 Planen eines Softwareproduktes

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung (AO 2020)

Teil 2 der Abschlussprüfung

4 Aufgaben
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Hinweis:
Bei der Bearbeitung der Aufgaben ist von einem gewöhnlichen Geschäftsbetrieb auszugehen, der **nicht** durch die COVID-19-Pandemie beeinflusst bzw. durch entsprechende behördliche Verfügungen eingeschränkt ist.

Bearbeitungshinweise

1. Bevor Sie mit der Bearbeitung der Aufgaben beginnen, überprüfen Sie bitte die **Vollständigkeit** dieses Aufgabensatzes. Die Anzahl der zu bearbeitenden Aufgaben ist auf dem Deckblatt links angegeben. Wenden Sie sich bei Unstimmigkeiten sofort an die Aufsicht, weil Reklamationen am Ende der Prüfung nicht anerkannt werden können.
2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
5. Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgaben** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
7. Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
10. Für **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor der Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

Wird vom Korrektor ausgefüllt!

Bewertung

Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen.

1. Aufg.		Punkte	2. Aufg.		Punkte	3. Aufg.		Punkte	4. Aufg.		Punkte
	15 16			17 18			19 20			21 22	

Prüfungszeit [ ] (23)
Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

Gesamtpunktzahl [ ] [ ] [ ] (24 25 26)

Prüfungsort, Datum

Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2021 – Alle Rechte vorbehalten!

Die Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:

Sie arbeiten im Projekt zur Entwicklung einer Software zur Terminvergabe in Arztpraxen mit. Dabei gehört es zu Ihren Aufgaben, bei der Planung Sicherheitsaspekte einzubeziehen sowie Teststrategien herauszuarbeiten.

Korrekturrand

1. Aufgabe (20 Punkte)

Bei der Vorbereitung des Projekts will das Projektteam grundlegende Fragen hinsichtlich der Programmierumgebung klären.

a) Zur Auswahl stehen Compiler- und Interpretersprachen. Sie sollen dem Projektteam die grundlegenden Unterschiede verdeutlichen.

aa) Nennen Sie je eine Programmiersprache für diese Kategorien. 2 Punkte

ab) Beschreiben Sie den Unterschied zwischen diesen Kategorien. 2 Punkte

ac) Nennen Sie jeweils einen Vorteil und einen Nachteil für die Verwendung eines Interpreters gegenüber einem Compiler. 2 Punkte

b) Ihr Entwicklungsteam hat sich für eine Programmiersprache entschieden. Für diese gibt es eine umfassende Klassenbibliothek, die Sie im Projekt auch einsetzen sollen.

ba) Beschreiben Sie einen Vorteil und einen Nachteil, der sich durch den Einsatz einer Bibliothek ergibt. 4 Punkte

bb) Sie planen, die Bibliothek um eigene Klassen zu erweitern.

Beschreiben Sie drei Aspekte, die Sie in der Dokumentation dieser Erweiterung beachten müssen. 6 Punkte

Korrekturrand

c) Ihr Entwicklungsteam besteht aus zehn Personen. Das Team schlägt vor, bei der Entwicklung mit einer Versionsverwaltung zu arbeiten.

Entscheiden Sie sich, ob Sie dem Vorschlag zustimmen, und geben Sie eine fachliche Begründung. 4 Punkte

bitte wenden!

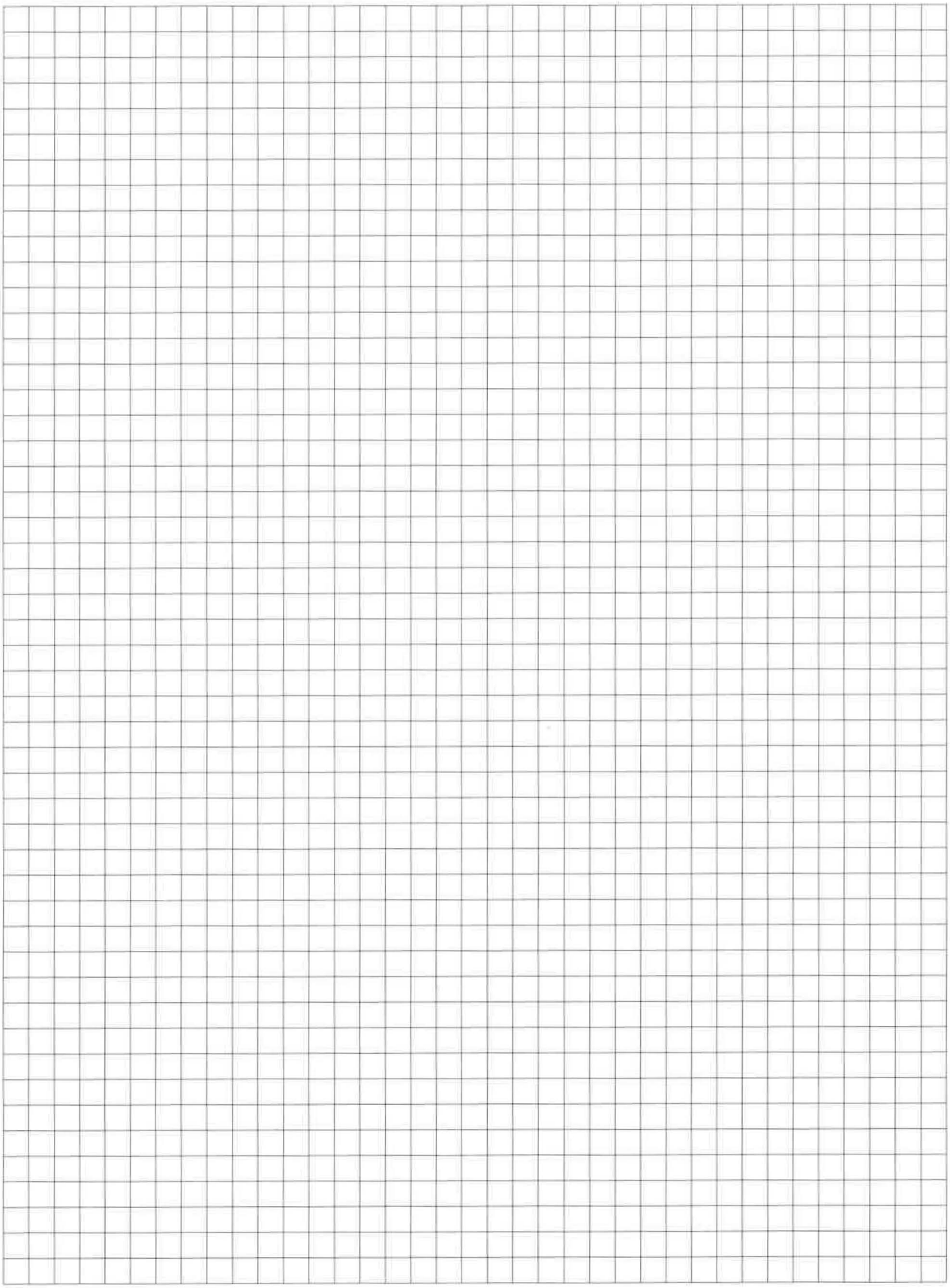
2. Aufgabe (30 Punkte)

Korrekturrand

a) Im Zuge des Projektes sollen Sie ein Anwendungsfalldiagramm erstellen, um eine übersichtliche Darstellung für das gesamte Team zu erhalten. Bei der Analyse ergeben sich folgende Anforderungen:

Mit dem Buchungssystem können Patienten und Angestellte der Ärzte arbeiten. Für das Buchungssystem ist ein vorheriger Login erforderlich. Patienten können Termine anfragen. Patienten können bei Bedarf aus einer Liste eine spezielle Terminart (Röntgen, EKG ...) auswählen. Ein Angestellter muss eine Terminanfrage abschließen (bestätigen oder absagen). Hierzu muss dieser sich zuerst eine Liste aller Terminanfragen anzeigen lassen. Bei dem Abschließen der Terminanfrage muss der Angestellte, wenn eine spezielle Terminart ausgewählt wurde, die Ressourcen planen.

Erstellen Sie ein Anwendungsfalldiagramm anhand der beschriebenen Anforderungen. 14 Punkte



b) Für die Terminverwaltung soll ein Datenbankmodell verwendet werden. Dazu soll zuerst die Terminbuchung dargestellt werden. Folgende Situationsbeschreibung liegt Ihnen vor.

In einer Gemeinschaftspraxis arbeiten Ärzte, von denen jeder die Betreuung mehrerer Patienten übernimmt. Dabei hat jeder Patient einen Arzt, der ihn für gewöhnlich betreut. Ärzte können sich gegenseitig vertreten. Für eine Vertretung wird das Start- und das Enddatum erfasst.

Ein Patient kann mehrere Termine vereinbaren, wobei zu einem Termin immer genau ein Patient zugeordnet wird. Zu einem geplanten Termin eines Patienten wird ein beliebiger Arzt zugewiesen. Generell kann ein Arzt aber mehrere Termine zugewiesen bekommen. Zu einem Termin wird die benötigte Terminart gespeichert. Eine Terminart kann mehreren Terminen zugewiesen werden. Zu jeder Terminart werden die entsprechenden Ressourcen hinterlegt. Dabei kann eine Ressource bei mehreren Terminarten ausgewählt werden.

Erstellen Sie ein ER-Modell für die Gemeinschaftspraxis. Verzichten Sie auf weitere Attribute außer den genannten. 16 Punkte

Korrekturrand

3. Aufgabe (25 Punkte)

Korrekturrand

Die Webapplikation für die Terminbuchung soll nun entwickelt werden.

a) Wesentliche Kriterien für die Akzeptanz einer Website sind Usability und positive User Experience.

aa) Verdeutlichen Sie anhand von vier verschiedenen Punkten, was darunter zu verstehen ist. 8 Punkte

ab) Die Benutzung soll außerdem barrierefrei möglich sein.
Erläutern Sie den Begriff. 4 Punkte

ac) Im Verlauf der Entwicklung soll die Güte der Usability periodisch überprüft und verbessert werden.
Beschreiben Sie zwei Möglichkeiten zum Test der Benutzerfreundlichkeit. 4 Punkte

b) Für den Login eines Benutzers *(User)* der Website ist das Klassenmodell wie folgt beschrieben:

- Der Benutzer meldet sich mit E-Mail-Adresse *(email)* und Passwort *(password)* an.
- Das eingegebene Passwort wird verschlüsselt *(encryptPassword)* und zurückgegeben.
- Das verschlüsselte Passwort wird mit dem in der Datenbank hinterlegten verglichen *(validatePassword)*. Bei Übereinstimmung wird true zurückgegeben.

Erstellen Sie das entsprechende UML-Klassendiagramm. Verwenden Sie die vorgegebenen kursiv gedruckten Bezeichner und geben Sie passende Datentypen an. 7 Punkte

Korrekturrand

c) Beim Test der Applikation wird im Browser *HTTP Error 404* angezeigt.

Beschreiben Sie eine mögliche Fehlerursache. 2 Punkte

4. Aufgabe (25 Punkte)

Korrekturrand

a) Um Daten zwischen der Client-Anwendung und dem Server nicht unverschlüsselt zu übertragen, wird vom Projektleiter vorgeschlagen, diese mit dem Lizenzschlüssel der Software zu verschlüsseln.

aa) Beschreiben Sie das oben dargestellte Verschlüsselungsverfahren. 2 Punkte

ab) Beschreiben Sie ein alternatives Verschlüsselungsverfahren mit jeweils einem Vor- und Nachteil, welches die zuvor genannten Probleme löst. 6 Punkte

b) Bei Ihrem Test der Client-Anwendung bekommen Sie beim Start die Meldung „Das Programm stammt aus einer nicht vertrauenswürdigen Quelle. Wollen Sie dieses Programm trotzdem öffnen?"

ba) Beschreiben Sie, was diese Meldung bedeutet. 2 Punkte

bb) Beschreiben Sie, wie Sie das Problem durch eine Änderung in den Kompilierungs-Einstellungen beheben können. 2 Punkte

Korrekturrand

c) Die Geschäftsführung Ihres Unternehmens beauftragt Sie, sich über Möglichkeiten zur Qualitätssicherung von Software zu informieren und Vorschläge zu unterbreiten.

ca) Bei Ihrer Recherche finden Sie die ISO Normen 9126 und 25010, welche Qualitätsmerkmale von Softwareanwendungen beschreiben. Zwei der darin beschriebenen Merkmale sind „Effizienz" und „Änderbarkeit".

Beschreiben Sie, was unter diesen beiden Qualitätsmerkmalen zu verstehen ist. 4 Punkte

cb) Aus Ihrem Projektteam kommt der Vorschlag, zur Qualitätssicherung ausschließlich automatische Code-Reviews und Unit-Tests einzusetzen.

Nehmen Sie Stellung zu diesem Vorschlag und begründen Sie fachlich Ihren Standpunkt. 5 Punkte

d) Sie schlagen Ihrem Projektteam vor, zusätzlich für jeden Release einen Black-Box- und White-Box-Test durchzuführen.

Beschreiben Sie diese beiden Testarten. 4 Punkte

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

[1] Sie hätte kürzer sein können.
[2] Sie war angemessen.
[3] Sie hätte länger sein müssen.

Abschlussprüfung Winter 2021/22

Belegsatz

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung (AO 2020)
1201

Teil 2 der Abschlussprüfung

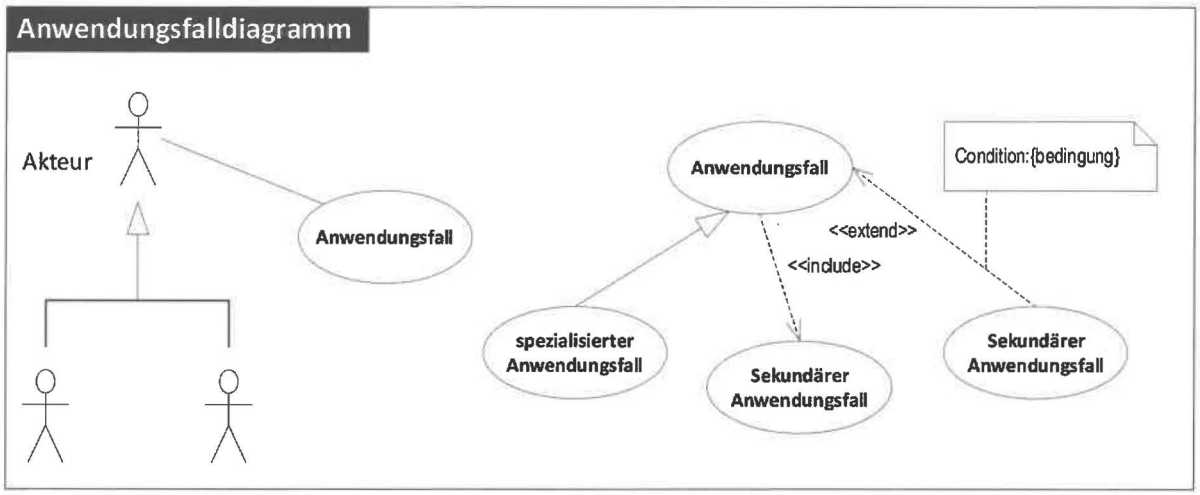
Der Belegsatz ist Grundlage für beide Prüfungsbereiche

1. Planen eines Softwareproduktes
2. Entwicklung und Umsetzung von Algorithmen

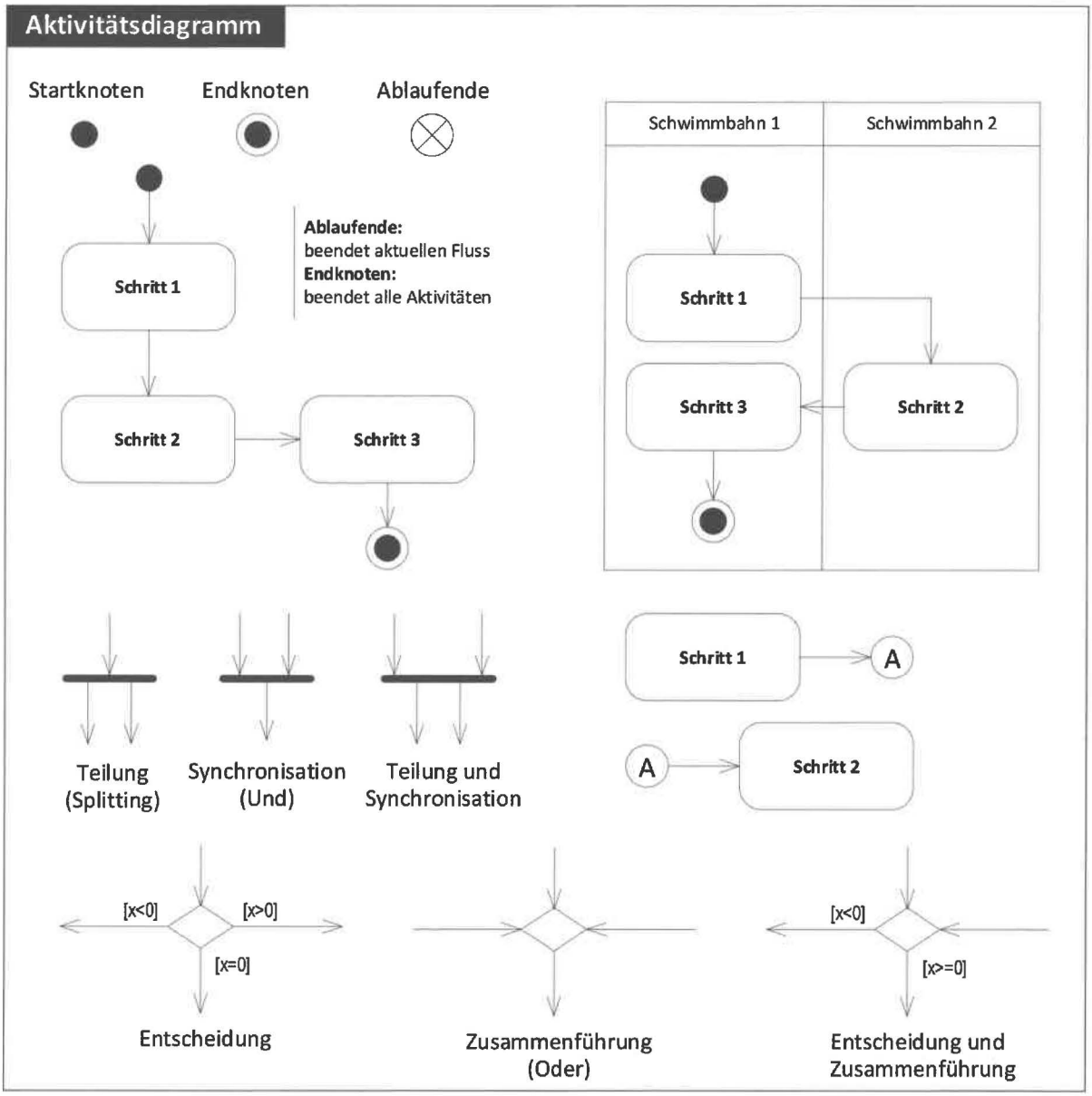
UML-Anwendungsfalldiagramm	2
UML-Aktivitätsdiagramm	2
UML-Klassendiagramm	3
SQL-Syntax (Auszug)	4/5

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2021 – Alle Rechte vorbehalten!

UML-Anwendungsfalldiagramm

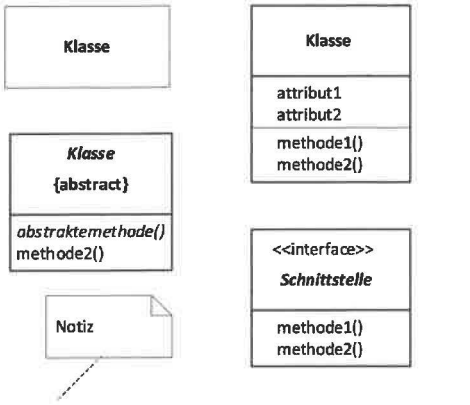


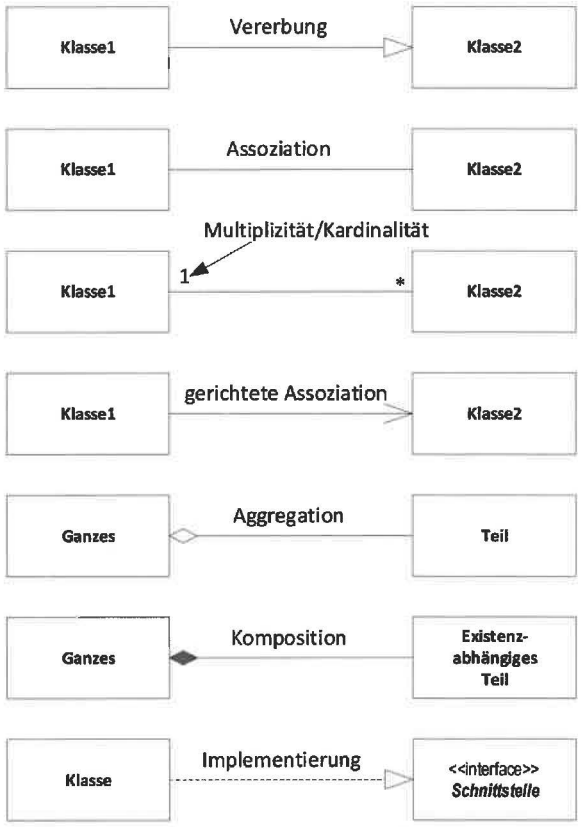
UML-Aktivitätsdiagramm



JML-Klassendiagramm

Klassendiagramm





Syntax für Attribute:
Sichtbarkeit Attributname : Typ {Eigenschaften}

Syntax für Methoden:
Sichtbarkeit Methodenname(parameter1 : Typ, ...) : Rückgabetyp {Eigenschaften}

Sichtbarkeit:
+ public
# protected
- private
~ package

Eigenschaften:
{static, final, ...}

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... (**SELECT** ... **FROM** ... **WHERE** ...) **AS** xyz **FROM** ... **WHERE** ...	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: *xyz)* ausgegeben.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) **WHERE NOT EXISTS** (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
GROUP BY Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, ...] **ASC** | **DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
Syntax	**Beschreibung**
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname **VALUES** (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder **SELECT** ... **FROM** ... **WHERE**	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind

Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(**time,**[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
NOT	Logische Negation
OR	Logisches ODER
IS	Test auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2018-03-29

Abschlussprüfung Winter 2021/22

Lösungshinweise

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung (AO 2020)
1201

1

Planen eines
Softwareproduktes

Teil 2 der Abschlussprüfung

Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.).

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1	=	100 – 92 Punkte	Note 2	= unter	92 – 81 Punkte
Note 3	= unter	81 – 67 Punkte	Note 4	= unter	67 – 50 Punkte
Note 5	= unter	50 – 30 Punkte	Note 6	= unter	30 – 0 Punkte

Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2021 – Alle Rechte vorbehalten!

1. Aufgabe (20 Punkte)

aa) 2 Punkte
Compilersprachen:
Z. B.: C, C++
Auch anzuerkennen sind Sprachen, die in Byte-Code kompiliert werden wie Java oder C#.

Interpretersprachen:
Z. B.: PHP, Python, BASIC

ab) 2 Punkte
Bei Compilersprachen muss zuerst eine Übersetzung durch den Compiler erfolgen. Nach erfolgter Übersetzung kann der erzeugte Code ohne erneute Übersetzung ausgeführt werden.

Bei Interpretersprachen übersetzt der Interpreter den Quellcode bei jeder Ausführung des Programms erneut.

ac) 2 Punkte
Vorteil eines Interpreters:
- Einfacher Entwicklungsprozess (Fehlerhafte Codestellen werden sofort gefunden.)
- Schnelles Testen ist möglich

Nachteil eines Interpreters
- Ineffizienter Übersetzungsprozess
- Langsamere Ausführungsgeschwindigkeit

ba) 4 Punkte
Vorteil:
In der Bibliothek sind Funktionen für bestimmte Aufgaben vorhanden, die dann nicht mehr neu entwickelt werden müssen.
Wenn eine Bibliotheksfunktion verändert wird, muss die erstellte Anwendung nicht verändert werden.

Nachteil:
Die Mitglieder des Projektteams müssen sich zunächst in die Bibliothek einarbeiten und benötigen wegen des Umfangs zusätzliche Zeit.

Overhead kann zu längerer Laufzeit führen.

bb) 6 Punkte
Hier sollen drei elementare Anforderungen an die Dokumentation einer Klasse beschrieben werden, wie z. B.:
- Zusammenfassende Beschreibung der Klasse
- Auflistung der Konstruktoren und Methoden
- Auflistung der von anderen Klassen geerbten Methoden
- Benennung der implementierten Interfaces
- Beschreibung jeder Methode im Detail durch Angabe von Datentyp und Bedeutung der Parameter, Datentyp und Bedeutung des Rückgabewertes sowie der Wirkung der Methode
- Benennung von Voraussetzungen für die Anwendung der Methoden

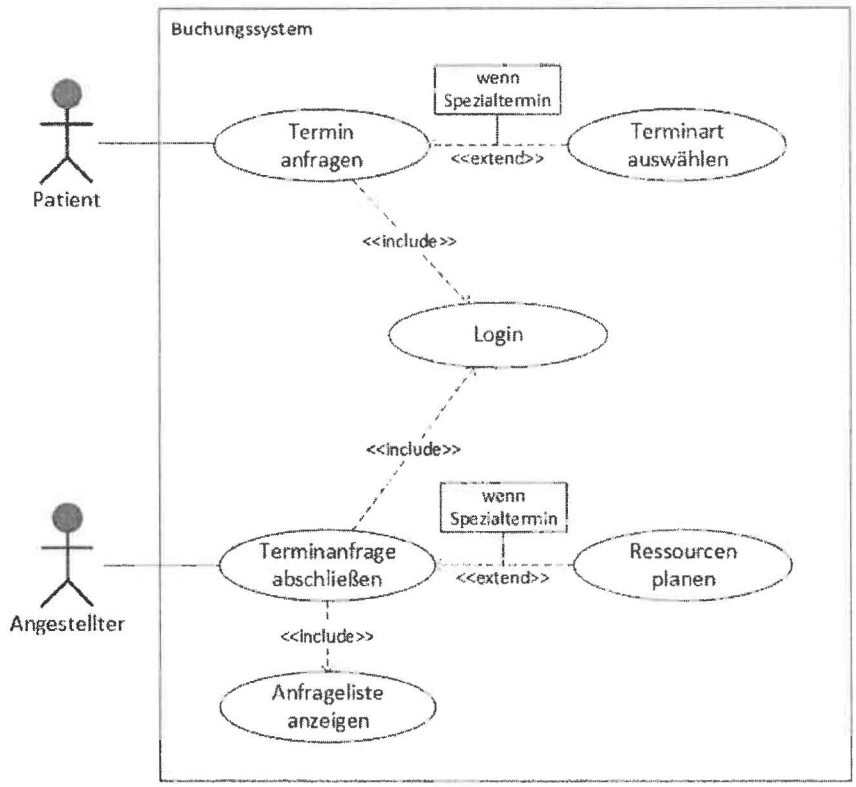
c) 4 Punkte
Dem Vorschlag ist zuzustimmen, da bei der angegebenen Größe des Entwicklungsteams individuelle Absprachen und Hinweise nicht ausreichen. Wegen des parallelen Arbeitens am System kann es bei einzelnen Programmteilen zu widersprüchlichen Änderungen kommen, die ohne Versionsverwaltung nicht rückgängig zu machen sind.

(Zustimmung 1 Punkt, Begründung 3 Punkte)

Andere sinnvolle Begründungen sind möglich.

2. Aufgabe (30 Punkte)

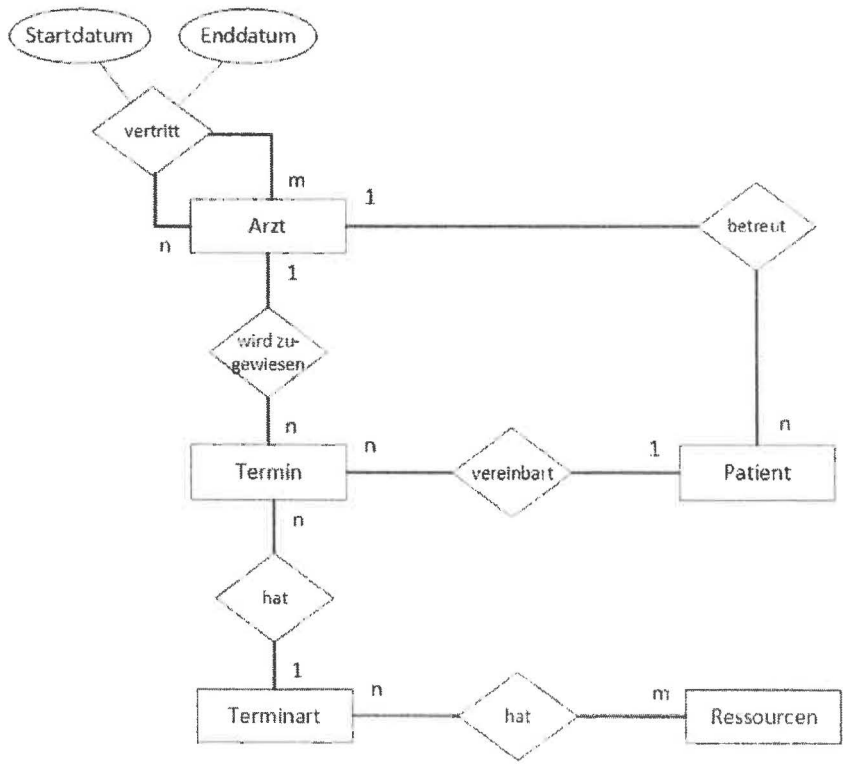
a) 14 Punkte



Andere Lösungen sind möglich.

1 Punkt Systemgrenze und Systemname
2 x je 1 Punkt pro Akteur mit geeigneter Rolle und Beziehung zum Anwendungsfall
6 x je 1 Punkt pro Anwendungsfall
5 x je 1 Punkt pro include- bzw. extend-Beziehung

b) 16 Punkte



Andere Lösungen sind möglich.

5 x je 1 Punkt pro Entitätstyp
6 x je 1,5 Punkte pro Beziehung und Kardinalität
2 x je 1 Punkt pro Attribut an der n:m-Beziehung

3. Aufgabe (25 Punkte)

aa) 8 Punkte (4 x je 2 Punkte)

- Die Navigation muss effizient und übersichtlich angelegt sein, z.B. Logo oben links, anklickbare Elemente sind als solche erkennbar.
- Der Nutzer muss die Funktionsweise schnell verstehen können.
- Praktische Funktionen wie Autovervollständigung sollen verwendet werden.
- Die Website muss auf unterschiedlichen Endgeräten gut benutzbar bleiben (responsive design).
- Das Erscheinungsbild muss attraktiv sein, evtl. CD-Vorgaben beachten.
- Der Nutzer muss das, was er sucht, schnell finden können.
- Kurze Ladezeiten für neue Contents.

ab) 4 Punkte für z. B.

Barrierefreiheit bedeutet, dass Menschen mit Behinderungen Webinhalte wahrnehmen und verstehen, in ihnen navigieren, mit ihnen interagieren und Beiträge zu ihnen leisten können. Websites, Tools und Technologien sind so zu gestalten, dass alle Menschen, einschließlich solcher mit Behinderungen, sie so einfach wie möglich verwenden können.

ac) 4 Punkte (2 x je 2 Punkte) für z. B.

- Direkte Rückmeldung durch Testuser mittels Online-Fragebogen oder Benutzertagebuch.
- Beobachtung von Testusern durch Testleiter oder automatisierte Tools, die Mausklicks, Navigationspfade etc. dokumentieren.
- Blickverlaufsmessungen (Eye-Tracking), lautes Denken der Testuser.
- Auswertung von Logfiles (Web Controlling), wer sieht was wie lange an.
- Erstellen von Anwendungsfällen, Card Sorting, Personas etc.

b) 7 Punkte

User	1 Punkt
email : String password : String	1 Punkt 1 Punkt
encryptPassword(String) : String validatePassword(String) : Boolean	2 Punkte 2 Punkte

c) 2 Punkte z. B. für

- Die URL bzw. deren Inhalte (wie Dateien oder Bilder) wurden entweder gelöscht oder ausgelagert (ohne dass die internen Links der Website hieran angepasst wurden).
- Die URL wurde fehlerhaft angelegt (etwa durch mangelnde Sorgfalt beim Anlegen oder dem Redesign), falsch verlinkt (z. B. durch eine inkorrekte Auszeichnung) oder in der Browserleiste falsch eingegeben.
- Der für die Webseite zuständige Webserver ist nicht in Betrieb oder die Verbindung ist unterbrochen.
- Der aufgerufene Domain-Name kann im Domain-Name-System nicht in eine IP umgewandelt werden.
- Den aufgerufenen Domain-Namen gibt es nicht (mehr).

4. Aufgabe (25 Punkte)

a) In den folgenden Lösungen werden Beschreibungen erwartet, wie z. B. die Nachfolgenden.
Alternative Lösungen (auch für diese konkrete Situation geltende) sind möglich.

aa) 2 Punkte
Bei dem symmetrischen Verschlüsselungsverfahren werden Daten mithilfe des gleichen Schlüssels ver- und entschlüsselt.

ab) 6 Punkte
Alternative Möglichkeiten können das „asymmetrische" und das „hybride Verschlüsselungsverfahren" sein.

Asymmetrische Verschlüsselung:
Hierbei kommt ein Schlüsselpaar zum Einsatz. Dieses setzt sich aus einem privaten und einem öffentlichen Schlüssel zusammen. Mit dem privaten Schlüssel wird entschlüsselt, mit dem öffentlichen verschlüsselt.

Hybride Verschlüsselung:
Ist die Kombination von symmetrischer und asymmetrischer Verschlüsselung. Ziel ist es, die Vorteile beider Systeme zu vereinen.

Asymmetrische Verschlüsselung:
+ Beide Clients können nur die für sie bestimmten Daten lesen
+ Einfache Realisierung der elektronischen Unterschrift
– Langsam und ineffizient
– Vertrauenswürdige Stelle wird benötigt, um die Authentizität des Senders/Empfängers zu verifizieren

Hybride Verschlüsselung:
+ Pre-Shared-Key für symmetrisches Verfahren wird je Verbindung dynamisch generiert/Clients müssen sich vorher nicht kennen
+ Effizient, da nur einmal eine kleine Datenmenge asymmetrisch verschlüsselt werden muss
– Produziert (verglichen mit den anderen beiden Verfahren) mehr Overhead beim Handshake für die Übertragung
– Vertrauenswürdige Stelle wird benötigt

Andere Lösungen sind möglich.

ba) 2 Punkte
Das Programm wurde beim Kompilieren nicht signiert.

bb) 2 Punkte
Um das Problem zu beheben, müssen beim Kompilieren die Dateien signiert werden. Das hierfür zu verwendende Zertifikat wird üblicherweise vom Hersteller des Betriebssystems ausgestellt.

ca) 4 Punkte
Effizienz:
Die erwartete Ausgabe des Programms soll mit möglichst wenig Rechenleistung erreicht werden. Je effizienter die Software arbeitet, umso größer ist die Bandbreite an möglicher Hardwareunterstützung (bezogen auf Hardwareressourcen).

Änderbarkeit:
Änderbarkeit ist die Möglichkeit, die Software durch individuelle Programmierung zu verändern. Je strukturierter eine Software aufgebaut ist, umso einfacher ist die spätere Änderbarkeit.

cb) 5 Punkte
Der Prüfling soll bei dieser Aufgabe erkennen, dass die ausschließliche Nutzung der genannten Testverfahren nicht alle Szenarien abdeckt und üblicherweise weitere (auch manuelle Tests) erforderlich sind, um die Fehlerfreiheit sicherzustellen. Insbesondere sollen in der Lösung Vor-/Nachteile/Gefahren benannt werden. Eine berechtigte Stellungnahme des Prüflings hinsichtlich Vorteilen/Nachteilen/Gefahren ist individuell zu bewerten.

d) 4 Punkte
Black-Box-Test:
Der Black-Box-Test, Funktionstest, erfordert keine Kenntnis über die innere Funktion der Software. Dieser Test wird von Anwendern durchgeführt, welche an der Entwicklung nicht beteiligt waren.

White-Box-Test:
Der White-Box-Test setzt Kenntnisse über den inneren Aufbau und die innere Funktion der Software voraus. Dieser Test wird vom Entwicklungsteam durchgeführt.